Assignment 3 Accidents

sai mithra

2023-10-15

**QUESTION1**

##Our goal here is to predict whether an accident just reported will involve an injury (MAX\_SEV\_IR = 1 or 2) or will not (MAX\_SEV\_IR = 0). ##For this purpose, create a dummy variable called INJURY that takes the value “yes” if MAX\_SEV\_IR = 1 or 2, and otherwise “no.”

#load the complete data set and install/load required packages.  
  
library(readr)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(e1071)

Accidents\_Data<- read.csv("C:\\Users\\saimi\\Downloads\\accidentsFull.csv")  
  
View(Accidents\_Data)

#create and insert a dummy variable called injury in the data.  
  
Accidents\_Data$INJURY <- ifelse(Accidents\_Data$MAX\_SEV\_IR>0, "yes", "no")  
  
for (i in 1:dim(Accidents\_Data)[2]) {  
 if (is.character(Accidents\_Data[, i])) {  
 Accidents\_Data[, i] <- as.factor(Accidents\_Data[, i])  
 }  
}  
head(Accidents\_Data, n=24)

## HOUR\_I\_R ALCHL\_I ALIGN\_I STRATUM\_R WRK\_ZONE WKDY\_I\_R INT\_HWY LGTCON\_I\_R  
## 1 0 2 2 1 0 1 0 3  
## 2 1 2 1 0 0 1 1 3  
## 3 1 2 1 0 0 1 0 3  
## 4 1 2 1 1 0 0 0 3  
## 5 1 1 1 0 0 1 0 3  
## 6 1 2 1 1 0 1 0 3  
## 7 1 2 1 0 0 1 1 3  
## 8 1 2 1 1 0 1 0 3  
## 9 1 2 1 1 0 1 0 3  
## 10 0 2 1 0 0 0 0 3  
## 11 1 2 1 0 0 1 0 3  
## 12 1 2 1 1 0 1 0 3  
## 13 1 2 1 1 0 1 0 3  
## 14 1 2 2 0 0 1 0 3  
## 15 1 2 2 1 0 1 0 3  
## 16 1 2 2 1 0 1 0 3  
## 17 1 2 1 1 0 1 0 3  
## 18 1 2 1 1 0 0 0 3  
## 19 1 2 1 1 0 1 0 3  
## 20 1 2 1 0 0 1 0 3  
## 21 1 2 1 1 0 1 0 3  
## 22 1 2 2 0 0 1 0 3  
## 23 1 2 1 0 0 1 0 3  
## 24 1 2 1 1 0 1 9 3  
## MANCOL\_I\_R PED\_ACC\_R RELJCT\_I\_R REL\_RWY\_R PROFIL\_I\_R SPD\_LIM SUR\_COND  
## 1 0 0 1 0 1 40 4  
## 2 2 0 1 1 1 70 4  
## 3 2 0 1 1 1 35 4  
## 4 2 0 1 1 1 35 4  
## 5 2 0 0 1 1 25 4  
## 6 0 0 1 0 1 70 4  
## 7 0 0 0 0 1 70 4  
## 8 0 0 0 0 1 35 4  
## 9 0 0 1 0 1 30 4  
## 10 0 0 1 0 1 25 4  
## 11 0 0 0 0 1 55 4  
## 12 2 0 0 1 1 40 4  
## 13 1 0 0 1 1 40 4  
## 14 0 0 0 0 1 25 4  
## 15 0 0 0 0 1 35 4  
## 16 0 0 0 0 1 45 4  
## 17 0 0 0 0 1 20 4  
## 18 0 0 0 0 1 50 4  
## 19 0 0 0 0 1 55 4  
## 20 0 0 1 1 1 55 4  
## 21 0 0 1 0 0 45 4  
## 22 0 0 1 0 0 65 4  
## 23 0 0 0 0 0 65 4  
## 24 2 0 1 1 0 55 4  
## TRAF\_CON\_R TRAF\_WAY VEH\_INVL WEATHER\_R INJURY\_CRASH NO\_INJ\_I PRPTYDMG\_CRASH  
## 1 0 3 1 1 1 1 0  
## 2 0 3 2 2 0 0 1  
## 3 1 2 2 2 0 0 1  
## 4 1 2 2 1 0 0 1  
## 5 0 2 3 1 0 0 1  
## 6 0 2 1 2 1 1 0  
## 7 0 2 1 2 0 0 1  
## 8 0 1 1 1 1 1 0  
## 9 0 1 1 2 0 0 1  
## 10 0 1 1 2 0 0 1  
## 11 0 1 1 2 0 0 1  
## 12 2 1 2 1 0 0 1  
## 13 0 1 4 1 1 2 0  
## 14 0 1 1 1 0 0 1  
## 15 0 1 1 1 1 1 0  
## 16 0 1 1 1 1 1 0  
## 17 0 1 1 2 0 0 1  
## 18 0 1 1 2 0 0 1  
## 19 0 1 1 2 0 0 1  
## 20 0 1 1 2 0 0 1  
## 21 0 3 1 1 1 1 0  
## 22 0 3 1 1 0 0 1  
## 23 2 2 1 2 1 2 0  
## 24 0 2 2 2 1 1 0  
## FATALITIES MAX\_SEV\_IR INJURY  
## 1 0 1 yes  
## 2 0 0 no  
## 3 0 0 no  
## 4 0 0 no  
## 5 0 0 no  
## 6 0 1 yes  
## 7 0 0 no  
## 8 0 1 yes  
## 9 0 0 no  
## 10 0 0 no  
## 11 0 0 no  
## 12 0 0 no  
## 13 0 1 yes  
## 14 0 0 no  
## 15 0 1 yes  
## 16 0 1 yes  
## 17 0 0 no  
## 18 0 0 no  
## 19 0 0 no  
## 20 0 0 no  
## 21 0 1 yes  
## 22 0 0 no  
## 23 0 1 yes  
## 24 0 1 yes

**QUESTION-1**

#Using the information in this dataset, if an accident has just been reported and no further information is available, what should the prediction be? (INJURY = Yes or No?) Why?

#CREATING A TABLE BASED ON INJURY.  
  
Injury\_Table <- table(Accidents\_Data$INJURY)  
show(Injury\_Table)

##   
## no yes   
## 20721 21462

#CALUCATING THE PROBABILITY OF INJURY   
  
Injury\_Probablilty =   
scales::percent(Injury\_Table["yes"]/(Injury\_Table["yes"]+Injury\_Table["no"]),  
0.01)  
Injury\_Probablilty

## yes   
## "50.88%"

**QUESTION-2**

#Select the first 24 records in the dataset and look only at the response (INJURY) and the two predictors WEATHER\_R and TRAF\_CON\_R.

#make a new subset with just the necessary records in it.  
  
Accidents\_Data24 <- Accidents\_Data[1:24, c('INJURY','WEATHER\_R','TRAF\_CON\_R')]   
Accidents\_Data24

## INJURY WEATHER\_R TRAF\_CON\_R  
## 1 yes 1 0  
## 2 no 2 0  
## 3 no 2 1  
## 4 no 1 1  
## 5 no 1 0  
## 6 yes 2 0  
## 7 no 2 0  
## 8 yes 1 0  
## 9 no 2 0  
## 10 no 2 0  
## 11 no 2 0  
## 12 no 1 2  
## 13 yes 1 0  
## 14 no 1 0  
## 15 yes 1 0  
## 16 yes 1 0  
## 17 no 2 0  
## 18 no 2 0  
## 19 no 2 0  
## 20 no 2 0  
## 21 yes 1 0  
## 22 no 1 0  
## 23 yes 2 2  
## 24 yes 2 0

#Create a pivot table that examines INJURY as a function of the two predictors for these 24 records. Use all three variables in the pivot table as rows/columns.

dtpvt1 <- ftable(Accidents\_Data24)  
dtpvt2 <- ftable(Accidents\_Data24 [,-1])  
  
dtpvt1

## TRAF\_CON\_R 0 1 2  
## INJURY WEATHER\_R   
## no 1 3 1 1  
## 2 9 1 0  
## yes 1 6 0 0  
## 2 2 0 1

dtpvt2

## TRAF\_CON\_R 0 1 2  
## WEATHER\_R   
## 1 9 1 1  
## 2 11 1 1

**Question-2(1)**

#Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors.

#QUESTION4  
#COMPUTING THE BAYES CONDITIONAL PROBABLITIES OF AN INJURY (INJURY = Yes) GIVEN THE SIX POSSIBILE COMBINATIONS OF THE PREDITCTORS.  
  
# Injury = yes  
  
Prob1 = dtpvt1[3,1] / dtpvt2[1,1] # Injury, Weather=1 and Traf=0  
Prob2 = dtpvt1[4,1] / dtpvt2[2,1] # Injury, Weather=2, Traf=0  
Prob3 = dtpvt1[3,2] / dtpvt2[1,2] # Injury, W=1, T=1  
Prob4 = dtpvt1[4,2] / dtpvt2[2,2] # I, W=2,T=1  
Prob5 = dtpvt1[3,3] / dtpvt2[1,3] # I, W=1,T=2  
Prob6 = dtpvt1[4,3]/ dtpvt2[2,3] #I,W=2,T=2  
print(c(Prob1,Prob2,Prob3,Prob4,Prob5,Prob6))

## [1] 0.6666667 0.1818182 0.0000000 0.0000000 0.0000000 1.0000000

# Injury = no  
  
N1 = dtpvt1[1,1] / dtpvt2[1,1] # Weather=1 and Traf=0  
N2 = dtpvt1[2,1] / dtpvt2[2,1] # Weather=2, Traf=0  
N3 = dtpvt1[1,2] / dtpvt2[1,2] # W=1, T=1  
N4 = dtpvt1[2,2] / dtpvt2[2,2] # W=2,T=1  
N5 = dtpvt1[1,3] / dtpvt2[1,3] # W=1,T=2  
N6 = dtpvt1[2,3] / dtpvt2[2,3] # W=2,T=2  
print(c(N1,N2,N3,N4,N5,N6))

## [1] 0.3333333 0.8181818 1.0000000 1.0000000 1.0000000 0.0000000

**QUESTION-2(2)**

#CLASSIFYING THE 24 ACCIDENTS USING THESES PROBABLITIES AND CUTOFF OF 0.5 #ADDING PROBABILITY RESULTS TO THE SUBSET

prob.inj <- rep(0,24)  
for (i in 1:24) {  
 print(c(Accidents\_Data24$WEATHER\_R[i],Accidents\_Data24$TRAF\_CON\_R[i]))  
 if (Accidents\_Data24$WEATHER\_R[i] == "1") {  
 if (Accidents\_Data24$TRAF\_CON\_R[i]=="0"){  
 prob.inj[i] = Prob1  
 }  
 else if (Accidents\_Data24$TRAF\_CON\_R[i]=="1") {  
 prob.inj[i] = Prob3  
 }  
 else if (Accidents\_Data24$TRAF\_CON\_R[i]=="2") {  
 prob.inj[i] = Prob5  
 }  
 }  
 else {  
 if (Accidents\_Data24$TRAF\_CON\_R[i]=="0"){  
 prob.inj[i] = Prob2  
 }  
 else if (Accidents\_Data24$TRAF\_CON\_R[i]=="1") {  
 prob.inj[i] = Prob4  
 }  
 else if (Accidents\_Data24$TRAF\_CON\_R[i]=="2") {  
 prob.inj[i] = Prob6  
 }  
 }  
}

## [1] 1 0  
## [1] 2 0  
## [1] 2 1  
## [1] 1 1  
## [1] 1 0  
## [1] 2 0  
## [1] 2 0  
## [1] 1 0  
## [1] 2 0  
## [1] 2 0  
## [1] 2 0  
## [1] 1 2  
## [1] 1 0  
## [1] 1 0  
## [1] 1 0  
## [1] 1 0  
## [1] 2 0  
## [1] 2 0  
## [1] 2 0  
## [1] 2 0  
## [1] 1 0  
## [1] 1 0  
## [1] 2 2  
## [1] 2 0

Accidents\_Data24$prob.inj <- prob.inj  
Accidents\_Data24$pred.prob <- ifelse(Accidents\_Data24$prob.inj>0.5, "yes", "no")  
table(Accidents\_Data24$pred.prob)

##   
## no yes   
## 14 10

**QUESTION-2(3)**

#COMPUTING MANUALLY THE NAIVE BAYES CONDITIONAL PROBABILITY OF AN INJURY GIVEN THE WEATHER\_R =1 AND TRAF\_CON\_R =1.

#The Naive Bayes conditional probability is computed using the Naive Bayes formula as follows: #P(INJURY = Yes | WEATHER\_R = 1 and TRAF\_CON\_R = 1) = (P(INJURY = Yes | WEATHER\_R = 1) \* P(INJURY = Yes | TRAF\_CON\_R = 1) \* P(INJURY = Yes)) / (P(WEATHER\_R = 1) \* P(TRAF\_CON\_R = 1))

Manual\_NB\_W1\_T1 <- Prob3  
cat("Manual Naive Bayes Conditional Probability (Injury = Yes | Weather\_R =   
1, TRAF\_CON\_R = 1):", Manual\_NB\_W1\_T1)

## Manual Naive Bayes Conditional Probability (Injury = Yes | Weather\_R =   
## 1, TRAF\_CON\_R = 1): 0

**QUESTION-3(4)**

#RUNNING A NAIVE BAYES CLASSIFIER ON THE 24 RECORDS AND TWO PREDICTORS. #NOW,WE HAVE TO CHECK THE MODEL OUTPUT TO OBTAIN PROBABILITIES AND CLASSIFCATIONS FOR ALL 24 RECORDS. ##AND THEN, WE ARE COMPARING TO BAYES CLASSIFCATION TO SEE IF THE RESULTING CLASSIFICATIONS ARE EQUIVALENT OR NOT.

library(e1071)  
  
NB<-naiveBayes(INJURY ~ ., data = Accidents\_Data24)  
  
NBT <- predict(NB, newdata = Accidents\_Data24,type = "raw")  
  
Accidents\_Data24$nbpred.prob <- NBT[,2] # Transfer the "Yes" nb prediction  
library(caret)  
  
NB2 <- train(INJURY ~ TRAF\_CON\_R + WEATHER\_R,   
 data = Accidents\_Data24, method = "nb")

## Warning: model fit failed for Resample01: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R

## Warning: model fit failed for Resample02: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R

## Warning: model fit failed for Resample07: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R, WEATHER\_R

## Warning: model fit failed for Resample08: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R, WEATHER\_R

## Warning: model fit failed for Resample09: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R

## Warning: model fit failed for Resample12: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R

## Warning: model fit failed for Resample13: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R

## Warning: model fit failed for Resample16: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R

## Warning: model fit failed for Resample22: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R

## Warning: model fit failed for Resample23: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R

## Warning: model fit failed for Resample24: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R

## Warning: model fit failed for Resample25: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R, WEATHER\_R

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,  
## : There were missing values in resampled performance measures.

predict(NB2, newdata = Accidents\_Data24[,c("INJURY", "WEATHER\_R", "TRAF\_CON\_R")])

## [1] yes no no yes yes no no yes no no no yes yes yes yes yes no no no   
## [20] no yes yes no no   
## Levels: no yes

predict(NB2, newdata = Accidents\_Data24[,c("INJURY", "WEATHER\_R", "TRAF\_CON\_R")],  
 type = "raw")

## [1] yes no no yes yes no no yes no no no yes yes yes yes yes no no no   
## [20] no yes yes no no   
## Levels: no yes

**QUESTION-3**

#Let us now return to the entire dataset. Partition the data into training (60%) and validation (40%).

#Splitting the data into training (60%) and validation (40%)  
  
set.seed(123)  
TrainIndex <- createDataPartition(Accidents\_Data$INJURY, p = 0.6, list =  
FALSE)  
Train\_Data <- Accidents\_Data[TrainIndex, ]  
Val\_Data <- Accidents\_Data[-TrainIndex, ]

**QUESTION-3(1)**

Run a naive Bayes classifier on the complete training set with the relevant predictors (and INJURY as the response). Note that all predictors are categorical. Show the confusion matrix.

#Creating a naive bayes model with the relavant predictors  
nb <- naiveBayes(INJURY ~ WEATHER\_R + TRAF\_CON\_R, data = Train\_Data)  
#Predicting on the validation set  
Val\_Pred <-predict(nb, newdata = Val\_Data)  
  
#Converting val\_pred into a character vector  
Val\_Pred <- as.character(Val\_Pred)  
  
#Converting val\_data$Injury to a character vector  
Val\_Data$INJURY <- as.character(Val\_Data$INJURY)  
  
#Creating a factor with matching levels  
Val\_Pred <- factor(Val\_Pred, levels = c("No", "Yes"))  
Val\_Data$INJURY <- factor(Val\_Data$INJURY, levels = c("No", "Yes"))  
  
#Creating a confusion matrix  
Confusion.Matrix <- confusionMatrix(Val\_Pred, Val\_Data$INJURY)  
print(Confusion.Matrix)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 0 0  
## Yes 0 0  
##   
## Accuracy : NaN   
## 95% CI : (NA, NA)  
## No Information Rate : NA   
## P-Value [Acc > NIR] : NA   
##   
## Kappa : NaN   
##   
## Mcnemar's Test P-Value : NA   
##   
## Sensitivity : NA   
## Specificity : NA   
## Pos Pred Value : NA   
## Neg Pred Value : NA   
## Prevalence : NaN   
## Detection Rate : NaN   
## Detection Prevalence : NaN   
## Balanced Accuracy : NA   
##   
## 'Positive' Class : No   
##

**QUESTION-3(2)**

#OVERALL ERROR OF THE VALIDATION SET   
  
Overall\_Error <- 1 - Confusion.Matrix$overall["Accuracy"]  
cat("overall error of the validation set:", Overall\_Error, "\n")

## overall error of the validation set: NaN